

LINGUISTIC SUMMARIES IN EVALUATING ELEMENTARY CONDITIONS AND THEIR APPLICATION IN SOFTWARE ENVIRONMENT

Pavol Sojka¹

Abstract: Data users are generally interested in two types of aggregated information: summarization of the selected attribute(s) for all considered entities and retrieval and evaluation of entities by the requirements posed on the relevant attributes. Less statistically literate users (e.g., domain experts) and the business intelligence strategic dashboards can benefit from linguistic summarization, i.e. a summary like most customers are middle-aged can be understood immediately. Evaluation of the mandatory and optional requirements of the structure P1 and most of the other posed predicates should be satisfied beneficial for analytical business intelligence dashboards and search engines in general. This work formalizes the integration of the aforementioned quantified summaries and quantified evaluation into the concept of database queries to empower their flexibility by, e.g., the nested quantified query conditions on hierarchical data structures. Later in our work, we adapted our research into practical application. We created a software environment for evaluating data based on a dataset retrieved from The Statistical Office of the Slovak republic. These datasets are aimed mainly on landscape characteristics like altitude, area sizes of towns and villages, and similar parameters. Based on user's preferences, our system recommends the most suitable place for holidays to spend on.

UDC Classification: 519.6, **DOI:** <https://doi.org/10.12955/pns.v2.159>

Keywords: fuzzy logic, linguistic summaries, degree of membership, computational intelligence, web application, databases

Introduction

Based on our previous research (Sojka et al., 2020), we decided to apply our work to the real world scenario described later in this paper. Datasets in databases usually contain a large number of entities and their attributes. Formally, a dataset can be expressed as a set of pairs (Skowron et al., 2015):

$$T_p = (U_p, A_p) \quad (1),$$

Where, U_p is a universe of entities (records) and A_p is a set of attributes in a table $T_p, p=1, \dots, n$. In such tables, rows are labeled by entities and columns by attributes. Generally, data users are interested in two types of queries, which might be expressed as vertical and horizontal aggregations (Rakovská and Hudec, 2019). In the former, statistical functions, such as means, deviations and distributions are used to explain the entities' attributes considering the set of entities U (1), e.g., the average altitude of all municipalities is m , whereas the standard deviation is s , where $m, s \in R$. In the latter, users are interested in entities that best satisfy the compound predicate, i.e., in finding the best entities considering the predicates posed on a subset of attributes A (1), such as *altitude BETWEEN (1000, 1200) and pollution ≤ 50 and population density < 100 and number of sunny days ≥ 120 and percentage of arable land ≥ 20* . In such queries, users have to express requirements by numbers, even though uncertainties about the borderline cases might appear (Keefe, 2000). In the third case, nested sub-queries posed against the 1 : N relationships (e.g., between the relations district and municipality) may require merging horizontal and vertical aggregations, such as select districts, where *unemployment rate ≥ 30 and percentage of respiratory diseases > 25* and the average pollution in their respective municipalities is higher than the limit value. For the data users, the natural way to express the requirements is by linguistic terms. The same holds for interpreting the summaries where, despite the broad use of statistical functions, they are suitable for domain experts having a certain level of statistical literacy (Hudec et al., 2018).

The literature has already recognized the limitations of the classical or two-valued approaches and provided solutions for the various situations. The vertical aggregation has been empowered by the so-called linguistic summaries initially proposed by Yager (Kacprzyk et al., 2002) and emphasized (Boran et al., 2016) that summaries should not be as terse as means. Since then, the theory of linguistically summarized sentences has been extensively researched by many scholars and applied in a variety of fields. A detailed, although not a very recent review, can be found in Boran et al. (2016). Less statistically literate users (e.g., domain experts and the general public) can benefit from such a summarization (Hudec et al., 2018; Schield, 2011). Through this approach, we are able to provide an overall overview of one attribute or relations among several attributes in a dataset, such as about half of the municipalities have the population density around the mean value, or the majority of young

¹ Ing. Pavol Sojka, University of Economics in Bratislava, Faculty of Economic Informatics, Department of Applied Informatics, e-mail: pavol.sojka@euba.sk

customers buy items in late evenings. Such summaries might improve the informativeness of business intelligence strategic dashboards, for instance.

A query against the data stored in a database provides a formal description of the entities of interest to the user posing this query (Hudec and Vučetić, 2015; Kacprzyk et al., 2000). Limitation of the two-valued logic in the database query conditions has been mitigated by the fuzzy query approaches like (Tamani et al., 2011; Hudec, 2009; Kacprzyk and Zadrożny, 2001; Wang et al., 2007). In this way, the most relevant entities with respect to user needs are retrieved together with their matching degrees, i.e. the closeness to the full satisfaction. An example of such a query is select customers having a high number of orders and low payment delay. Next, a user might be interested in entities that meet the majority of requirements. Such a query is of the structure the most of atomic requirements $\{P_1 \dots P_n\}$ should be met (Kacprzyk and Zadrożny, 2013). However, this approach is not able to make a distinction between the mandatory and optional requirements. Further, linguistic summaries have shown their applicability as nested subqueries in the hierarchical data structures, e.g., select regions where most municipalities meet the requirement P (Hudec, 2016). More complex nested queries require the integration of vertical and horizontal aggregations. The foundation for all the aforementioned approaches is the theory of fuzzy sets introduced by Zadeh (Dubois and Prade, 2012), the theory of fuzzy logic based on the theories of many-valued logics and fuzzy sets, and the theory of aggregation functions summarized in (Dubois and Prade, 2004), (Beliakov et al. 2007). Thus, the methodology of our work is based on the key findings in these fields. The research questions in this work are the following: the problem of merging the horizontal and vertical aggregation and the formalization of mandatory and optional predicates in quantified queries, and a subsequent proposal of a suitable integration. By this approach, we can cover the gap in the merging of quantified summarization with evaluation. In addition, when a conjunctively expressed query condition consists of a larger number of predicates, an empty answer might appear. The proposed aggregation by the fuzzy quantifier *most of* is a semantically different contribution than the existing approaches covering the empty answer problem (Bosc et al., 2009, 2008, 2007; Smits et al., 2014), and therefore it augments the established ones.

Linguistic Summaries in Brief

This section studies the relevant theoretical aspects of data summaries by short quantified sentences of natural language. A basic structure of such sentence has the form Q entities in a dataset: P where Q is a linguistic quantifier such as *most of*, *about half* and *few*, and P is an elementary or compound predicate. The truth value (or validity) is calculated in the following way (Yager, 1982):

$$v(Q_x(P(x))) = \mu_Q\left(\frac{1}{n} \sum_{i=1}^n \mu_P(x_i)\right) \quad (2)$$

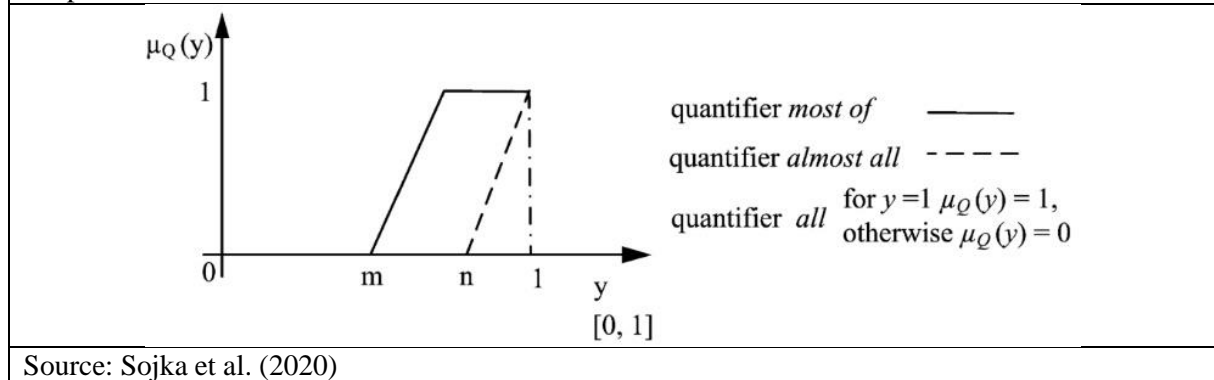
where n is the number of entities or the scalar cardinality of a dataset (a universe of entities U_p (1)), $\frac{1}{n} \sum_{i=1}^n \mu_P(x_i)$ is the proportion of entities in a dataset that satisfies predicate P , $\mu_P(x_i)$ is the matching degree of entity x_i to predicate P , and μ_Q is the membership function of a chosen relative quantifier. The truth value v assumes values from the unit interval.

Formalization of fuzzy relative quantifiers can be carried out by using three methods: sigma-counts (Farnadi et al., 2014), Ordered Weighted Averaging (OWA) operator (Yager and Kacprzyk, 2012), and Competitive Type Aggregation (Xu and Zhou, 2011). The sigma-count method is adopted for this work because it allows the quantifiers and predicates to be modeled in the same way, which simplifies the applicability, and therefore is more intuitive for diverse users. Within this method, the quantifier *most of* is formalized by an increasing (usually linear) function. It can be constructed independently by equations offered in Kacprzyk and Zadrożny (2013) or as one granule from the family of uniformly distributed relative quantifiers constructed on the $[0,1]$ interval (Hudec, 2016). When expressed by parameters, the quantifier *most of* yields (see, Figure 1):

$$\mu_Q(y) = \begin{cases} 1 & y \geq n \\ \frac{y-n}{n-m} & m < y < n \\ 0 & y \leq m \end{cases} \quad (3)$$

where $0.5 \leq m \leq n \leq 1$. When $m=n=1$, the quantifier becomes the crisp quantifier *all*, whereas when, e.g., $0.8 \leq m < n=1$, the quantifier expresses the term *almost all*.

Figure 1: Parametrized linguistic quantifier most of, where y is the proportion of entities that meet the predicate P



Evaluation of Atomic Conditions by Quantified Summaries

This section studies linguistic summaries employed as aggregations of elementary requirements in the evaluation of entities.

Evaluation of Optional Atomic Conditions

In database queries, the usual way of selecting the relevant entities is realized via the conditions expressed by a conjunction (AND operator) and disjunction (OR operator). The former cannot cover the aggregation of mandatory and optional requirements because all the requirements are mandatory. If only one atomic condition from a larger set is rejected, the overall matching degree is zero (zero is the absorbing element in conjunction). The latter is based on the substitutability principle, i.e., one satisfied atomic requirement is sufficient.

Let us recall the standard classification of aggregation functions (Dubois and Prade, 2004). Conjunctive aggregation functions are characterized by $A(x) \leq \min(x)$, disjunctive by $A(x) \geq \max(x)$, averaging by $\min(x) \leq A(x) \leq \max(x)$, and remaining aggregation functions are called mixed, where x is a vector, $x = (x_1, \dots, x_n)$. Thus, the following problems in conjunctive aggregation might appear. First, mandatory and optional requirements, were addressed by the asymmetric *AND IF POSSIBLE* conjunction (Dujmović, 2007; Bosc and Pivert, 2012) and axiomatized in Hudec and Mesiar (2020). Second, the aforementioned empty answer problem, i.e., cases when not a single record meets a larger set of atomic conditions. Third, all atomic predicates might be optional where the principle the more, the better holds, i.e., an entity is preferred over another one if it satisfies more predicates. Thus, to exclude weakly performing entities, and on the other side to mitigate the empty answer problem, conjunction is relaxed by the quantifier *most of*, that is, most of the atomic predicates should be satisfied. The query relaxation by the fuzzy relative quantifier: *the most of atomic conditions should be satisfied* is initially suggested by Kacprzyk and Zadrozny (2013). It is formalized by the quantified summaries (2), (3) as follows:

$$v(x) = \mu_Q\left(\frac{1}{n} \sum_{i=1}^n \mu_{P_i}(x)\right) \quad (4)$$

where n is the number of atomic predicates posed on a subset of attributes A (1), $\frac{1}{n} \sum_{i=1}^n \mu_{P_i}(x)$ is the proportion of atomic predicates P_i that are satisfied by the entity x being evaluated and μ_Q is the formalization of the quantifier *most of*. The truth value v assumes values from the unit interval.

Evaluation of Mandatory and Optional Atomic Conditions

In many cases, several atomic requirements are mandatory, while the other ones are optional, and moreover, if a higher proportion of optional requirements is satisfied, then the entity is more suitable. In order to cover this aggregation requirement, we have modified equation (4) in the following way

$$v(x) = \bigwedge_{i=1}^r P_i(x) \wedge \mu_Q\left(\frac{1}{s} \sum_{j=1}^s \mu_{P_j}(x)\right) \quad (5)$$

where r is the number of mandatory requirements and s is the number of optional requirements (usually $r < s$), and x is an evaluated entity.

A suitable method for formalizing *AND* connective (conjunction) in fuzzy logic is by triangular norms (or in short t-norms), because of the desirable properties (monotonicity, associativity, symmetry, and the presence of a neutral element) (Hájek, 2013). The four basic t-norms are (Klement et al., 2005):

minimum t-norm, product t-norm, Łukasiewicz t-norm and drastic product. The least suitable is drastic product due to its very restrictive nature and non-continuity. The product t-norm and Łukasiewicz t-norm have a downward reinforcement property, whereas the minimum t-norm has the property of idempotency (Beliakov et al., 2007). For instance, when using the Łukasiewicz t-norm, the solution is greater than zero when both mandatory and quantified parts are significantly satisfied. We have two conjunctions in equation (5) – among mandatory and between mandatory and quantified requirements. In addition, there exist conjunctive functions which do not meet all the axioms of t-norms, e.g. $C(a,b)=av \cdot bw$ (Beliakov et al., 2007; Hudec and Vučetić, 2019), where $v > 1$ and $w > 1$ indicate the importance of predicates. Observe that for $v=w=1$ we get the product t-norm and for $v < 1$, $w < 1$ and $v+w=1$ we get the geometric mean. Although the other functions could be examined, this work is focused on the minimum t-norm.

Integrating the model into an application environment

When creating models, it is often criticized that many models are less tested and thus not usable in application practice. Authors often try a model on theoretical and small data samples. We tested our model first on a small scale to reveal and remove shortcomings that are easier to reveal in smaller data samples. After testing validity data in Microsoft Excel, we switched to implementing the model in the program environment. Our environment includes using the Oracle’s MySQL database server, it is also possible to use the Open Source version (Open Source - Open source software, often with free spreading and distribution) MariaDB, which is developed on the basis of MySQL server, and therefore, these two database systems are compatible. As a presentation layer we used Apache Web server from Apache Foundation with PHP scripting language. The user enters an address into the internet browser and then loads the page content where the users can insert their files containing data in CSV (Comma Separated Value) format. Subsequently, our system automatically processes data and creates all the desired structures in the database on the fly. The advantage of this approach is that our system can dynamically process the variable number of input attributes. Thus, the user need not to manually define table structures in the database, namely the number of columns and their names, which is a relevant property for users not familiar with databases and tables creation processes and thus our system advantage. The system will allow users to work with these data after loading. For the simplicity of the end-user who is not expected to have a deeper knowledge of methods of statistical data processing, fuzzy logic, or other mathematical models, the interface is easy, intuitive and a minimum interaction is required on each of several application’s screens. On the following screen, the user selects which parameters from the inserted data he/she is interested in. Then continues to the next part of the application, where he/she selects the value of the previously selected parameters with the slider. The system indicates the minimum, maximum, and average parameter’s value for the basic orientation in given values. The user may or may not change parameters values. When values stay unchanged, the system will use predefined ones.

Figure 2: Fine-tuning of selected attributes

Municipality_altitude	
Min: 100	<input type="text" value="308"/>
Max: 993	
Avg: 308	<input type="text" value="0.8"/>
	<input type="text" value="1.2"/>

Source: Author

The system automatically generates so-called dynamic SQL queries whose fine-tuning was relatively difficult, because this type of query creation may occur, in special cases, hardly predictable errors. After the calculation is started, the user continues to the last screen, where the system shows the resulting

values after the calculation is finished. We put into a database data obtained from The Statistical Office of the Slovak Republic, which describes the individual municipalities in Slovakia in terms of their location according to altitude, area, and proportion of agricultural land area on the municipality's total area. After evaluating the data, the result (Figure 3) is returned and shows which individual municipalities fall into our preferential selection with what intensity. The municipalities are then linked to Google's maps, which is an added value, user after clicking on the name of the municipality, immediately sees where it is located. Other features can also be implemented into such interfaces, which, for example, shows the basic information about the municipality, the possibilities of spending free time around, potential cultural and sports options, and others. Thanks to the web page hyperlinks of other systems like Wikipedia, this is already possible and allows you to find some additional information on linked websites.

Figure 3: Two uppermost records retrieved from the database (municipalities)

Dolný Lieskov	1.00
Kotmanová	1.00

Source: Author

Dynamically generated SQL query is as convenient because it is not fixed on column names (attributes, predicates) of the database table, so user can use this tool as a universal library to evaluate records and in the final step results can be interpreted accordingly as shown on the Figure 3. Excerpt from the code Generating SQL command follows:

```
$sql="{ $brace[$key]} `{$cols_names[$key]}` >= {$nums[$key]}*{$low[$key]} {$logical[$key]} ";
```

In this way, the formation of SQL command passes several steps until the final form is reached to be sent to perform a database query. This method of forming an SQL query is quite challenging because a detailed accuracy when creating variables, parentheses, operators, and necessary components of such a query are needed. The implementation of the mathematically formalized query is realized as a function of PHP language called when needed for calculating the intensity degree.

As we have seen from the example above, this approach can be applied to the mid or lower management layer in the company involved in developing and constructing buildings intended for individual houses for living or holiday house purposes, and this tool can ease to make decisions to find suitable places across the country by defining, e.g., nearby terrain characteristics.

Conclusion

In our data society, we face the problems of diverse needs for the aggregation of atomic requirements for the evaluation of entities and for explaining summarized information. In order to contribute, we raised the research question of merging the quantified evaluation and the quantified summarization, as well as aggregating the mandatory and optional quantified predicates in a fuzzy environment, where the requirements are expressed via fuzzy sets.

The answer is that the quantified evaluation (so-called horizontal quantified aggregation) and the data summarization (vertical aggregation) are supported by the fuzzy relative quantifiers and therefore can be straightforwardly merged to answer complex queries. In this work, the relative quantifiers are formalized by sigma-counts, and therefore, the quantifiers and predicates are modeled by the same method, which simplifies the applicability and is more intuitive for users.

For the quantified evaluation, we should consider cases where some of the atomic requirements are mandatory. In this case, we should aggregate mandatory and optional quantified requirements by a conjunctive function. The conjunction in our paper is realized by the minimum t-norm. Future research shall consider other conjunctive functions to cover various conjunctive aspects of the aggregation of mandatory and optional quantified requirements. The quantified aggregation of atomic predicates is a relaxation of conjunction, which augments the existing approaches to mitigate empty answer problems.

The suggested approach is demonstrated on examples in order to illustrate the diverse needs of the users. Real-world tasks such as flexible recommendations, informing and searching in smart cities, and business intelligence dashboards might benefit from the results of this work.

Acknowledgments

This work was supported by a project VEGA No. 1/0373/18 entitled "Big data analytics as a tool for increasing the competitiveness of enterprises and supporting informed decisions" by the Ministry of Education, Science, Research and Sport of the Slovak Republic.

References

- Beliakov, G., Pradera, A., Calvo Sánchez, T. (2007). *Aggregation Functions: A Guide for Practitioners*. Springer-Verlag, Berlin, Heidelberg.
- Boran, F.E., Akay, D., Yager, R.R. (2016). An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications*, 61, 356–377.
- Bosc, P., Pivert, O. (2012). On four noncommutative fuzzy connectives and their axiomatization. *Fuzzy Sets and Systems*, 202, 42–60.
- Bosc, P., Hadjali, A., Pivert, O. (2007). Weakening of fuzzy relational queries: and absolute proximity relation-based approach. *Mathware and Soft Computing*, 14, 35–55.
- Bosc, P., Hadjali, A., Pivert, O. (2008). Empty versus overabundant answers to flexible relational queries. *Fuzzy Sets and Systems*, 159, 1450–1467.
- Bosc, P., Brando, C., Hadjali, A., Jaudoin, H., Pivert, O. (2009). Semantic proximity between queries and the empty answer problem. In: *Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference*, Lisbon, pp. 259–264.
- Dubois, D., Prade, H. (2004). On the use of aggregation operations in information fusion processes. *Fuzzy Sets and Systems*, 142, 143–161.
- Dubois, D., Prade, H. (2012). *Fundamentals of Fuzzy Sets*. Springer Science & Business Media.
- Dujmović, J. (2007). Continuous Preference Logic for System Evaluation. In: *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1082–1099, Dec. 2007, doi: 10.1109/TFUZZ.2007.902041.
- Farnadi, G., Bach, S.H., Moens, M.F., Getoor, L. & De Cock, M. (2014). Extending PSL with fuzzy quantifiers. In: *Proceedings of the 13th AAAI Conference on Statistical Relational AI (AAAIWS'14-13)*. AAAI Press, 35–37.
- Hájek, P. (2013). *Meta Mathematics of Fuzzy Logic*. Springer-Science+Business media, B.V.
- Hudec, M. (2009). An approach to fuzzy database querying, analysis and realization. *Computer Science and Information Systems*, 6, 127–140.
- Hudec, M. (2016). *Fuzziness in Information Systems – How to Deal with Crisp and Fuzzy Data in Selection, Classification, and Summarization*. Springer, Cham.
- Hudec, M., Mesiar, R. (2020). The axiomatization of asymmetric disjunction and conjunction. *Information Fusion*, 53, 165–173.
- Hudec, M., Vučetić, M. (2015). Some issues of fuzzy querying in relational databases. *Kybernetika*, 51, 994–1022.
- Hudec, M., Vučetić, M. (2019). Aggregation of fuzzy conformances. In: *10th International Summer School on Aggregation Operators, AGOP 2019, Olomouc, Czech Republic*.
- Hudec, M., Bednářová, E., Holzinger, A. (2018). Augmenting statistical data dissemination by short quantified sentences of natural language. *Journal of Official Statistics*, 34, 981–1010.
- Kacprzyk, J., Pasi, G., Vojtáš, P., Zadrožny, S. (2000). Fuzzy querying: issues and perspectives. *Kybernetika*, 36, 605–616.
- Keefe, R. (2000). *Theories of Vagueness*. Cambridge University Press, Cambridge.
- Kacprzyk, J., Yager, R. R., & Zadrožny, S. (2002). Fuzzy Linguistic Summaries of Databases for an Efficient Business Data Analysis and Decision Support. In *Knowledge Discovery for Business Information Systems* (pp. 129–152). Kluwer Academic Publishers. https://doi.org/10.1007/0-306-46991-x_6
- Kacprzyk, J., Zadrožny, S. (2001). SQLf and FQUERY for Access. *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, 2001, pp. 2464–2469 vol.4, doi: 10.1109/NAFIPS.2001.944459.
- Kacprzyk, J., Zadrožny, S. (2013). Comprehensiveness of Linguistic Data Summaries: A Crucial Role of Protoforms. In *Computational Intelligence in Intelligent Data Analysis* (pp. 207–221). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-32378-2_14
- Kacprzyk, J., Zadrožny, S. (2013). Compound bipolar queries: combining bipolar queries and queries with fuzzy linguistic quantifiers. *Proceedings of the 8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13)*. <https://doi.org/10.2991/eusflat.2013.125>
- Klement, E.P., Mesiar, R., Pap, E. (2005). Triangular norms: basic notions and properties. In: Klement, E.P., Mesiar, R. (Eds.), *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms*. Elsevier, Amsterdam, pp. 17–60.
- Rakovská, E., & Hudec, M. (2019). Two approaches for the computational model for software usability in practice. *Advances in Intelligent Systems and Computing*, 191–202. doi:10.1007/978-3-030-18058-4_15
- Schild, M. (2011). Statistical literacy: a new mission for data producers. *Statistical Journal of the IAOS*, 27, 173–183.
- Skowron, A., Jankowski, A., Swiniarski, R.W. (2015). Foundations of rough sets. In: Kacprzyk, J., Pedrycz, W. (Eds.), *Handbook of Computational Intelligence*. Springer, Berlin, Heidelberg, pp. 331–348.

Smits, G., Pivert, O., Hadjali, A. (2014). Fuzzy cardinalities as a basis to cooperative answering. In: Pivert, O., Zadrozny, S. (Eds.), *Flexible Approaches in Data, Information and Knowledge Management. Studies in Computational Intelligence*, Vol. 497. Springer, Cham, pp. 261–289.

Sojka, P., Hudec, M., Švaňa, M. (2020). Linguistic Summaries in Evaluating Elementary Conditions, Summarizing Data and Managing Nested Queries. In *Informatica : An International Journal*. Vilnius : Vilnius University. Vol. 4, 31, 841-856.

Tamani N., Liétard L., Rocacher D. (2011) Bipolar SQLf: A Flexible Querying Language for Relational Databases. In: Christiansen H., De Tré G., Yazici A., Zadrozny S., Andreasen T., Larsen H.L. (eds) *Flexible Query Answering Systems. FQAS 2011. Lecture Notes in Computer Science*, vol 7022. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24764-4_41

Wang, T-C., Lee, L.H-D., Chen, C-M. (2007). Intelligent queries based on fuzzy set theory and SQL. In: 39th Joint Conference on Information Science, Salt Lake City, pp. 1426–1432.

Xu, J., Zhou, X. (2011). Fuzzy-Like Multiple Objective Decision Making. In *Studies in Fuzziness and Soft Computing*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-16895-6>

Yager, R.R., Kacprzyk, J. (2012). *The ordered weighted averaging operators: theory and applications*. Springer Science & Business Media.